



Application Development Aids



Burroughs

Technical Information Paper

DATE: April 3, 1980

Author: J. DiNapoli
MSC West
El Monte, CA

PRODUCT: LARGE SYSTEMS DMSII

SUBJECT: HOW TO INTERPRET LARGE SYSTEM DMSII STATISTICS

This document discusses the statistical report created by DMSII. The statistics report is generated, for databases with the STATISTICS option set, either at the time the last user of the database closes the database or by the SM STATISTICS message for the visible DBS on the ODT.

This report is a snapshot of all activity to the database from the last time the statistical variables were reset. The statistical variables are reset at the first open of the database and through the visible DBS command SM STATISTICS RESTART.

The database statistics report is divided into six sections:

- Time interval for the statistics report.
- Buffer statistics.
- Input/output statistics.
- Database usage statistics.
- Audit statistics.
- Transaction statistics.

TIME INTERVAL

This states the date and time that the database was opened and the start and stop time for the period in which statistics were gathered.

TO: BMG Communications Code RC

Printed in U.S.A.

1114238-014

Technical Information Paper

BUFFER STATISTICS

The ALLOWEDCORE figure is the setting currently in use by the accessroutines. This figure might be different from the figure stated in the DASDL compile listing if the visible DBS was used to change the original setting.

The MAXIMUM BUFFER STORAGE USED indicates the maximum amount of memory used to satisfy the DMSII buffer requirements for the database, for this statistical time interval.

The NUMBER OF BUFFERS USED is a variable that indicates the maximum number of buffers ever allocated during the statistics interval.

The NUMBER OF DATA BASE OVERLAYS indicates the number of times the accessroutines deallocated a buffer. Buffers are deallocated for the following reasons:

- A structure is closed.
- A user changes from serial mode to random mode.
- A user changes from random mode to serial mode.
- The amount of memory used for buffers (TOTALCORE) exceeds the DASDL specified amount (ALLOWEDCORE).

Care should be exercised to prevent an excessive overlay rate; however, this may result in the use of more memory. An overlay rate of about 1 per second for all databases concurrently active is considered to be reasonable, for low transaction rates only. An overlay rate near zero would be ideal. When a buffer is overlaid (meaning written back to disk and deallocated, or perhaps only deallocated) for one of the above reasons, all process stacks are searched for copy descriptors to the buffer. This can impact the entire system if many process stacks are present.

Technical Information Paper

The ALLOWEDCORE variable can be used to help keep the overlay rate down. Experimentation is the best technique for choosing a good allowedcore value. Set allowedcore to be absurdly high value (100 K words or more) and let the database seek its own buffer level. After the database has been running for some time, look at the MAXIMUM BUFFER STORAGE USED. Reset allowedcore to a value slightly lower than maximum buffer storage used. After the database has been running for a while longer, look at the NUMBER OF DATA BASE OVERLAYS that occurred for that period. If the value is reasonable, the allowedcore setting is suitable; otherwise, a larger value if required. (The choosing of the allowedcore value should be attempted only after the buffer requirements have been determined and tuned for a stable database environment.)

INPUT/OUTPUT STATISTICS

The next part of the report contains information about the physical I/O's against the database for each structure that has been opened at least once. If ABORT had run, all non-partitioned structures are opened by ABORT; therefore, all structures will be represented in the report.

The name of each structure is listed in a column on the left side of the report and information about I/O activity is listed in columns adjacent to the structure name. Because the description of each type of entry is too long to be used as a column heading, each column consists of an integer which refers to a corresponding head-note.

This section is divided into two parts; information pertaining to reads, followed by information pertaining to writes. Columns 2 through 5 are for reads, while columns 6 through 9 are for writes.

Technical Information Paper

The first column (#1) is the I/O TIME in seconds. This is the total I/O time used to access the records of the structure excluding seek time, but including latency and data transfer time. It is the file attribute F.IOCLOCKS in seconds.

The next column (#2) is the total NUMBER OF READS (NORMAL PLUS READ AHEAD) for the structure. It is incremented for each physical read. If the READAHEAD option is set for the structure, the third column, NUMBER OF READ AHEAD, will indicate the number of physical I/O's performed to pre-fetch a block in advance of its use. This number is also included in the figure in column 2. If the DASDL option READAHEAD is not set, zeroes will appear in column 3.

The next column (#4) is the total WAIT TIME FOR READS (IN SECONDS), this is the accumulated elapsed time for all programs waiting for physical reads to complete after all DMSII housekeeping has been completed.

What is meant by DMSII housekeeping is the following. After it has been determined by the accessroutines that the desired record is not in a buffer in memory, an I/O for the read is initiated. The request is processed by the MCP. Rather than having the accessroutines wait for the I/O to complete, the oldest buffer for the structure the read was initiated for is examined. If this buffer has been modified, and the changes have already been written to the audit trail then, the accessroutines initiate the write of this buffer. This process is known as WRITEAHEAD. If the I/O for the original read has not yet completed, the user will be put to sleep and the wait time for reads will start to accumulate. If the I/O for the original read had completed before the completion of these housekeeping activities, then there will be no wait time.

Technical Information Paper

When there is an abundance of housekeeping, the amount of wait time could be lower than the I/O time (column 1). However, wait time can include part or all of the I/O time, as well as possibly including seek time (which is not included in the I/O TIME) and time spent waiting in the I/O queue.

Column #5, AVERAGE WAIT TIME FOR READS (IN MILLISECONDS) is the quotient of dividing column #4 by column #2. A reasonable average without READAHEAD enabled in DASDL is about 50 to 60 milliseconds per read, on 225 or 235 packs. Some of the factors contributing to a higher average read wait time can be:

- Contention to the disk spindle.
- Contention to the pack control.
- Large mix with processor bound jobs.
- Excessively large block of data to be transferred.
- I/O error retries.

The following columns refer to physical writes to the database.

Column #6 is the total NUMBER OF WRITES (NORMAL PLUS WRITE AHEAD) for the structure. It is incremented for each physical write.

Column #7 is the NUMBER OF WRITE AHEADS for the structure. This is performed by the housekeeping process of the DMSII read.

The next column (#8) is the total WAIT TIME FOR WRITES (IN SECONDS). This is the elapsed time accumulated while a task is asleep waiting for a write to complete.

Technical Information Paper

The last column (#9) is the AVERAGE WAIT TIME FOR WRITES (IN MILLISECONDS), this is the quotient of dividing columns #8 by column #6. An average write wait time of about 15 to 25 milliseconds is considered to be reasonable. Some of the reasons that would account for a higher average write time are:

- The WRITEAHEAD function within DMSREAD is not occurring often enough.
- Too many buffers are being written by the controlpoint procedure.
- The overlay procedure is occurring too often.
- There is contention for the pack or the path to the pack.
- Too much I/O time is being spent in transferring large data blocks.

DATABASE USAGE STATISTICS

The statistics in this section indicate the number of DMSII statements performed against each structure. These statistics, in conjunction with the I/O STATISTICS, can help the database administrator and programmer determine possible design of programming problems. Similar to the I/O STATISTICS, the name of each structure is listed in a column to the left and column headings indicate the activity of various DMSII statements are reported.

There is no exception activity count for each structure listed. In many instances, the structures activity count is incremented prior to the exception check; however, this is not true for SECURITYERROR, READONLY, NOTLOCKED, and VERIFY DATAERROR exceptions

Technical Information Paper

AUDIT STATISTICS

This section of the statistics report will not appear for non-audited databases or databases opened INITIALIZE.

The STARTING and ENDING AUDIT FILE NUMBERS and the STARTING and ENDING AUDIT BLOCK SERIAL NUMBERS are indicated.

The AVERAGE BLOCK SIZE (IN WORDS) indicates the average number of words in an audit block compared to the maximum number of words that can fit in an audit block.

An audit block is written from one of two audit buffers in memory. A full audit block is written when the audit buffer is filled. A partially full audit block is written at sync-point time or when the amount of memory for present buffers exceed ALLOWEDCORE and the accessroutines need to reuse a modified buffer that cannot be written to disk (due to the fact that the audit information for the buffer has not been physically written to the audit media). When auditing to disk, the average will be close to the actual block size, because partially filled audit blocks are reused. The last audit block may be partially full, accounting for the average block size being slightly smaller than the actual block size. When auditing to tape, partially filled blocks are not reused, accounting for the average block size being smaller than the actual block size. If the average audit block size is much smaller than the actual block size either reduce the size of the audit block or increase the number of transactions that occur per sync-point.

The AVERAGE I/O WAIT column indicates the average wait time for writes to the audit file. When auditing to tape a reasonable wait time is about 3 to 8 millisecond. When auditing to pack, 10 to 15 milliseconds or less is good.

Technical Information Paper

If the average wait time is high, it is possible that the audit file is being written too frequently. The audit block size might be too small causing it to fill too quickly; therefore, requiring the full audit buffer to be written to the audit file. The syncpoint value might be too small causing the audit buffer to be flushed too often. The audit buffer can also be written to the audit file when TOTALCORE exceeds ALLOWEDCORE. If there is a high number of overlays, increasing the allowedcore value should reduce the number of overlays and the average I/O wait time on the audit file.

TRANSACTION STATISTICS

This section of the report will not appear for non-audited databases nor contain valid information for databases opened INITIALIZE.

TOTAL TRANSACTION COUNT is a variable that is incremented at each BEGIN-TRANSACTION. The value reflects the total number of times users entered transaction state. If the transaction is aborted and does not complete the transaction count is not decremented.

TOTAL SYNCPOINT COUNT is the total number of times syncpoint processing was done. It does not include syncpoints that cause controlpoint processing.

TOTAL CONTROLPOINT COUNT is the total number of times controlpoint processing was done.

WAIT FOR SYNCPOINT/CONTROLPOINT COUNT is the total number of times tasks were put to sleep waiting for either a syncpoint or a controlpoint to finish.

Technical Information Paper

When a task is put to sleep waiting for the completion of a syncpoint or a controlpoint its waiting time is accumulated. This accumulated time is divided by the wait for syncpoint/controlpoint count to produce AVERAGE WAIT FOR SYNCPOINT/CONTROLPOINT. Typically controlpoints take more time to complete than syncpoints; therefore, this average wait time can be severely influenced by controlpoint processing.

When the accessroutines start controlpoint processing, the time spent to complete the controlpoint is accumulated. This time is divided by the total controlpoint count to produce the AVERAGE TIME TO TAKE CONTROLPOINT. This includes the time to write the current audit buffer, write all modified buffers not flushed at the previous controlpoint, write storage control information, and update all disk file headers for structures whose LASTRECORD attribute has changed. A large number of buffers per structure can affect this time.

The AVERAGE NUMBER OF BUFFERS FLUSHED AT CONTROLPOINT is the next item, indicating the average number of modified buffers written to disk at a controlpoint.

PERCENTAGE OF MODIFIED BUFFERS FLUSHED AT CONTROLPOINT specifies the percent of all modified buffers that were written to disk, as a result of controlpoint processing. The best value for this statistic is 0%, indicating that all modified buffers were written to disk by the writeahead process, database close, or by the overlay procedure. By controlling the audit buffer size, the number of buffers per structure, the syncpoint frequency, and the controlpoint frequency this value can be kept low. 50% is the worst case, indicating that half of the modified buffers are written at controlpoint and modified buffers are not being written by WRITEAHEAD.